6

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/704,790	11/03/2000	Walter Mason Stewart	109993.00103	7495

27557 7590 12/18/2002

BLANK ROME COMISKY & MCCAULEY, LLP
900 17TH STREET, N.W., SUITE 1000
WASHINGTON, DC 20006

EXAMINER
KLIMACH, PAULA W

ART UNIT	PAPER NUMBER
2131	

DATE MAILED: 12/18/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

6

8

Office Action Summary	**Application No.** 09/704,790	**Applicant(s)** STEWART ET AL.	
	Examiner Paula W Klimach	**Art Unit** 2131	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on *02 August 2002* .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)□ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)□ Claim(s) *1-41* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

DETAILED ACTION

Drawings

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: 110. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 4, 16, 19 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al (5,832,208) in view of Templeton (6,401,210) and Newton's Telecom dictionary.

In reference to 1 and 16, Chen discloses a system for protecting the network from a virus contained in an email message, which has a server 20 that is used to transmit and receive files and email messages from and to other nodes. The definition of a gatekeeper provided by the Newton's Telecom dictionary is a device that is in charge of a gate. Therefore, a gatekeeper restricts items coming in and out of an entrance. Server 20 performs the task of the gatekeeper server, of receiving the e-mail messages, column 7 lines 4-10, and determines which files have viruses, and therefore, which files are allowed into the system and which ones are not, fig. 3.

The system disclosed by Chen can be used to carry out the method of claim 1 as mentioned above.

Chen does not disclose a system for converting the executable code from an executable format to a non-executable format.

Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. In one embodiment, Templeton gives the user the choice to direct that the infected filese be cleaned of a virus, restored to the original storage location without cleaning, deleted, saved to a different storage location and possibly renamed, or sent to another user while disabled or scrambled, column 3 lines 66 to column 4 line 4. Sending the files to another user includes forwarding the non-executable to the recipient of the e-mail message.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to receive the e-mail from the server 20, gatekeeper, then the user may direct that the infected file be cleaned or a virus, restored to the original storage location without cleaning, deleted, saved to a different storage location and possibly renamed, or sent to another user while disabled or scrambled, as directed by one embodiment of Templeton. The recipient of the message would be the other user described by Templeton. One of ordinary skill in the art would have been motivated to do this because deleting the file, as performed in the system described by Chen, would destroy the files original contents, Template column 1 lines 28-31. Destroying the files may entail considerable and possibly irreparable damage to a user's data, programs or file systems, column 1 lines 31-42.

In reference to Claim 31, Chen discloses a server 20 that is used to transmit and receive email messages and attachments from and to other nodes, column 7 lines 4-10. The email message is then sent to the corresponding node as shown in Fig 3.

Chen does not disclose a sacrificial server that converts the executable code from an executable format to a non-executable format. The definition of "sacrificial server" is not provided in the specification, as a result the examiner chooses to define a sacrificial server as a server that can be sacrificed. The server 20 is used only for the purpose of receiving email, detecting viruses, and managing the virus infected files, as a result the server can be sacrificed without any damage being inflicted on the other devices on the network.

Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. The method may be used to scramble the e-mail attachment and gives the user the choice to clean the file of the virus, restore to the file to the original storage location without cleaning, delete, save to a different storage location and possibly renamed, or sent to another user while disabled or scrambled.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to have a sacrificial server on the network, which comprises of communication means for receiving an e-mail attachment from the network; and processing means for converting the e-mail attachment from an executable format to a non-executable format; and processing means for returning the e-mail attachment to the network. One of ordinary skill in the art would have been motivated to do this because deleting the file, as performed in the system described by Chen, would destroy the files original contents, Template column 1 lines 28-31.

Destroying the files may entail considerable and possibly irreparable damage to a user's data, programs or file systems, column 1 lines 31-42.

In reference to claim 4 and 19, in addition to what was applied to claim 1 and 16, Chen further discloses a system where the executable code is contained in an attachment in the e-mail message, column 6 lines 63-67.

Claims 2 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al in view of Templeton and the Newton's Telecom dictionary as applied to claims 1 and 16 respectively above, and further in view of Cornetto et al and the Microsoft dictionary.

Chen does not describe executable code embedded email as files that can contain viruses.

Cornetto teaches of HTML formatted email, which acts like a browser, page 1 paragraph 4 and 5.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to receive the email using Chen's server 20, search for HTML formatted email, search for executables within the email as described by Cornetto (page 2 paragraph 1), and then the user may direct that the infected file be cleaned or a virus, restored to the original storage location without cleaning, deleted, saved to a different storage location and possibly renamed, or sent to another user while disabled or scrambled. One of ordinary skill in the art would have been motivated to do this because embedded executables run locally and if they contain malicious scripts they could create disruptive virus behavior, page 2 paragraph 1.

Claims 3 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al in view of Templeton, Newton's Telecom dictionary, Cornetto, and Microsoft Computer dictionary as applied to claim 2, and 17 respectively above, and further in view of Allen (5940614) and Brown.

Allen discloses a system that can deactivate and reactivate hyperlinks to provide a hypertext control method and apparatus in which different hypertext information or different target modules are displayed based upon a user class or authority, column 2 lines 2-6.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to receive email as in Chen server 20, look for email with links, deactivate hyperlinks as in Allen. One of ordinary skill in the art would have been motivated to do this because clicking on hyperlinks in email can lead to virus infection, Brown paragraph 20.

Claims 5, 6, 20, 21, and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton and Newton's Telecom dictionary as applied to claims 4, 16, and 31 above, and further in view of Schnurrer et al (5,842,002), Microsoft Computer dictionary.

In reference to claim 5, Chen discloses a system that forwards the email message to the server to the workstation column 7 lines 4-10. Templeton discloses a system where the executable code is converted to non-executable by scrambling the code, column 2 lines 45-48.

However, Chen and Templeton do not disclose a system where there exists a sacrificial server where virus activity is examined.

Schnurer et al discloses a device that can run the file with the virus (Fig 6B), detect a virus (Fig 6C), and recover from the effects of the virus (Fig 6C). The Microsoft computer dictionary defines a device as a generic term for a computer subsystem and as a result the Schnurer device may be used as a subsystem of a server, page 141.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the Schnurer device to Chen's server 20 to run the attachments described by Chen. One of ordinary skill in the art would have been motivated to do because the executable file would perform the disruptive behavior and the virus would be discovered, Schnurer column 4 lines 38-41.

In reference to claim 6, 21, and 32, Schnurer discloses a system that looks for computer virus activity which include changes in the IRQ table, FAT, and files, Fig. 6C.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to look for virus activity in server 20. One of ordinary skill in the art would have been motivated to do this because virus activity indicates the presence of a virus in the network, Schnurer column 7 lines 53-67.

In reference to claim 20, Chen does not disclose a sacrificial server, which is separate from the gatekeeper server.

Schnurer discloses a virus trap that is separate from the server as in fig. 4.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the Schnurer virus trap to a server 20 and set it apart from the gatekeeper server. One of ordinary skill in the art would have been motivated to do this because it would not inconvenience the user and is more effective because it guarantees a clean start, column 3 lines 20-23.

Claims 7, 22 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton, Newton's Telecom dictionary, Schnurer and the Microsoft

Computer dictionary as applied to claims 6, 21 and 32 respectively above, and further in view of Meyer et al (6,170,055).

In reference to claim 7 and 22 Chen, Templeton and Schnurer do not disclose a method wherein rebooting the sacrificial server is carried out from a safe copy of an operating system obtained from a read-only device.

Meyer discloses a computer recovery system where removable high capacity disk is used to reboot the system, as in the abstract. The high capacity removable media has ROM for storage of programs, column 9 lines 52-53.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to recover the sacrificial server 20 using a safe copy of the operating system stored in ROM as disclosed in Meyer. One of ordinary skill in the art would have been motivated to do this because the ability to have a clean uninfected start, Schnurer column 3 lines 20-23.

In reference to claim 33, Chen and Templeton do not disclose a server where there exists a sacrificial server, a server that comprises a read-only device and is rebooted from a safe copy of an operating system obtained from the read-only device.

Meyer discloses a computer recovery system where removable high capacity disk is used to reboot the system, as in the abstract. The high capacity removable media has ROM for storage of programs, column 9 lines 52-53.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to boot the server 20 from a safe copy of the operating system described by Meyer.

One of ordinary skill in the art would have been motivated to do this because the ability to have a clean uninfected start, Schnurer column 3 lines 20-23.

Claims 8 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton, Newton's Telecom dictionary, Schnurer, and the Microsoft Computer dictionary as applied to claims 5 and 20 respectively above, and further in view of Swift et al (6,377,691 B1).

Chen, Templeton and Schnurer do not disclose a method wherein communication between the gatekeeper server and the sacrificial server is authenticated using a challenge-and-response technique.

Swift discloses a system that uses a challenge-response authentication technique to authenticate the communication between a client and server, abstract.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to use a challenge-response authentication technique to authenticate the sacrificial server, 20. One of ordinary skill in the art would have been motivated to do this because the challenge-response authentication technique prevents the replay of messages therefore detecting intruders, column 3 lines 19-21 and column 3 lines 44-46.

Claims 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton and Newton's Telecom dictionary as applied to claim 31 above, and further in view of Swift.

Chen and Templeton do not disclose a method wherein communication between the gatekeeper server and the sacrificial server is authenticated using a challenge-and-response technique.

Swift discloses a system that uses a challenge-response authentication technique to authenticate the communication between a client and server, abstract.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to use a challenge-response authentication technique to authenticate the sacrificial server, 20. One of ordinary skill in the art would have been motivated to do this because the challenge-response authentication technique prevents the replay of messages therefore detecting intruders, column 3 lines 19-21 and column 3 lines 44-46.

Claims 9, 24, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton and Newton's Telecom dictionary as applied to claims 4, 16, and 31 respectively above, and further in view of Ji et al (5,623,600) and Battersby et al (5,740,370).

Chen and Templeton do not disclose a system that maintains a list of approved attachment types.

Battersby discloses a system that maintains a list of file type identifiers in order to determine whether a file belongs to a certain file subset of files, column 14 lines 18-25.

Ji discloses a system that determines whether the attachment is of a type, which is in the list of approved attachments types (types that do not contain viruses), fig 6B. Ji also sends a virus detection message to the client as a reply, fig 6B.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to maintain a list of approved attachment types in server 20, determine whether the attachment is a type which is in the list described by Battersby with the method described by Ji, and inform the recipient that a message containing a non-approved attachment has been received, as described by Ji. One of ordinary skill in the art would have been motivated to do this because it would not affect the performance of individual computers, column 2 lines 23-30.

Claims 10, 25, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton and Newton's Telecom dictionary as applied to claims 1, 16, and 31 respectively above, and further in view of Ji et al (5,623,600) and Jury et al (5,618,054).

In reference to 10 and 25, Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48, therefore deactivating the executable code.

Chen and Templeton do not disclose a system that maintains a list of approved executable code.

Jury discloses a process that maintains a list of files retrieved by the user in order to delete the files when the user terminates the Electronic Performance Support System, column 10 lines 17-22.

Ji discloses a system that determines whether the attachment is of a type, which is in the list of approved attachments types (types that do not contain viruses), fig 6B. Ji also sends a virus detection message to the client as a reply, fig 6B. The types of files that are suspected virus carriers are executable code, column 7 lines 33-40.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to maintain a list of approved executable code as described by Jury, determine whether the attachment is executable code in the list of executable code as in Ji, and deactivate the executable code if it is not in the list, as described by Templeton. One of ordinary skill in the art would have been motivated to do this because maintaining a list of files gives the user a choice of files to deactivate, as shown in column 7 lines 32-38.

In reference to 36, Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48, therefore deactivating the executable code.

Chen and Templeton do not disclose a server that maintains a list of approved executable code.

Jury discloses a process that maintains a list of files retrieved by the user in order to delete the files when the user terminates the Electronic Performance Support System, column 10 lines 17-22.

Ji discloses a system that determines whether the attachment is of a type, which is in the list of approved attachments types (types that do not contain viruses), fig 6B. Ji also sends a virus detection message to the client as a reply, fig 6B. The types of files that are suspected virus carriers are executable code, column 7 lines 33-40.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to maintain a list of approved executable code as in Jury, determine whether the attachment is executable code in the list of executable code as in Ji, and deactivate the executable code if it is not in the list, as described by Templeton. One of ordinary skill in the art would have been motivated to do this because maintaining a list of files gives the user a choice of files to deactivate, as shown in Jury column 7 lines 32-38.

Claims 11, 12, 13, 26, 27, 28, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton, Newton's Telecom dictionary, Ji, and Jury as applied to claims 10, 25, and 36 above, and further in view of Corthell (6,192,477 B1), Horwitt, and Rad.

In reference to claim 11, 12, 13, 26, 27, and 28 Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. This would deactivate the executable code.

Chen, Templeton, Ji and Jury do not disclose a method for determining whether the executable code has been altered, using an check-summing algorithmic technique.

Corthell discloses a system that uses the checksum algorithm to determine whether a file has been altered, column 8 lines 7-16.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered using the method of Corthell, using an algorithm. Then deactivate the executable code if it has been altered using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because checksum is a common way to check if files have been altered, Horwitt, abstract.

In reference to claim 13, Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. This would deactivate the executable code.

Chen, Templeton, Ji, and Jury do not disclose a method for determining whether the executable code has been altered, using a check-summing algorithmic technique.

Corthell discloses a system that uses the checksum to determine whether a file has been altered, column 8 lines 7-16.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered, using a checksum algorithm as shown in Corthell. Then deactivate the executable code if it has been altered using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because checksum is a common algorithm of checking the corruption of files, Horwitt, abstract. If the file has been corrupted (altered), it is possible that it contains a virus, Rad, paragraphs 13-16nad 30.

In reference to 37, 38, and 39, Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. This would deactivate the executable code.

Chen, Templeton, Ji and Jury do not disclose server for determining whether the executable code has been altered, using an check-summing algorithmic technique.

Corthell discloses a system that uses the checksum algorithm to determine whether a file has been altered, column 8 lines 7-16.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered, using a checksum algorithm, as shown by Corthell. Then deactivate the executable code if it has been altered, using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because checksum is a common algorithm of checking the corruption of files, Horwitt, abstract. If the file has been corrupted (altered), it is possible that it contains a virus, Rad, paragraphs 13-16nad 30.

Claims 38, and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton, Newton's Telecom dictionary, Schnurer, and the Microsoft Computer dictionary as applied to claim 32 above, and further in view of Corthell (6,192,477 B1), Horwitt, and Rad.

In reference to 38 and 39, Templeton discloses a method for scrambling the contents of a file thus rendering it unrecognizable or non-executable, column 2 lines 45-48. This would deactivate the executable code.

Chen, Templeton, and Schnurer do not disclose server for determining whether the executable code has been altered, using an check-summing algorithmic technique.

Corthell discloses a system that uses the checksum algorithm to determine whether a file has been altered, column 8 lines 7-16.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered, using a checksum algorithm, as shown by Corthell. Then deactivate the executable code if it has been altered, using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because checksum is a common algorithm of checking the corruption of files, Horwitt, abstract. If the file has been corrupted (altered), it is possible that it contains a virus, Rad, paragraphs 13-16nad 30.

Claims 14, 29, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton, Newton's Telecom dictionary, Ji, Jury, and Corthell, Horwitt, and

Rad as applied to claims 12, 27, and 38 respectively above, and further in view of Helbig Sr. et al (6,311,273 B1).

In reference to claim 14 and 29, Templeton, Ji, Jury, and Corthell do not disclose a method that utilizes a hashing function.

Helbig discloses the use of a hashing algorithm to determine if control software has been altered, column 1 lines 57-60.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered, using a hashing algorithm, as shown by Helbig. Then deactivate the executable code if it has been altered, using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because a hashing function is a secure method of determining that code has not changed and thus that the trusted code has not been altered, column 1 lines 65-68.

In reference to claim 40, Templeton, Ji, Jury, and Corthell do not disclose a server that utilizes a hashing function.

Helbig discloses the use of a hashing algorithm to determine if control software has been altered.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to determine whether the executable code has been altered, using a hashing algorithm, as shown by Helbig. Then deactivate the executable code if it has been altered, using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because a hashing function is a secure method of determining that code has not changed and thus that the trusted code has not been altered, column 1 lines 65-68.

Claims 15, 30, and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen in view of Templeton and Newton's Telecom dictionary as applied to claims 1, 16, and 31 above, and further in view of Field et al (6, 253, 324).

In reference to 15 and 30, Chen and Templeton do not disclose a method where a copy is make of the executable code, executing the first copy and not the second copy and comparing the effect of the executable code.

Field discloses a system where executable code is stored in an image file and the same code is copied and then executed in and executable image. Then a comparison is made of the non-writeable sections of the executable image and that of the verified image file. If the images match then the client is verified, column 2 lines 30-45.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to make a copy of the executable code in server 20 and run one copy and not the other in order to compare the result of running the code, as disclosed by Field. Then deactivate the executable code if it has been altered. One of ordinary skill in the art would have been motivated to do this because running one copy of the code and not the other a way to detect attacking programs that modify memory images of legitimate programs in order to alter its execution, column 2, lines 17-27.

In reference to claim 41, Chen and Templeton do not disclose a server that copies the executable code, executing the first copy and not the second copy and comparing the effect of the executable code.

Field discloses a system where executable code is stored in an image file and the same code is copied and then executed in and executable image. Then a comparison is made of the non-writeable sections of the executable image and that of the verified image file. If the images match then the client is verified, column 2 lines 30-45.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to make a copy of the executable code in server 20 and run one copy and not the other in order to compare the result of running the code, as disclosed by Field. Then deactivate the executable code if it has been altered, using the method disclosed by Templeton. One of ordinary skill in the art would have been motivated to do this because running one copy of the code and not the other a way to detect attacking programs that modify memory images of legitimate programs in order to alter its execution, column 2, lines 17-27.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6,401,210	Templeton
5,940,614	Allen et al
6,377,691 B1	Swift et al
6,311,273 B1	Helbig et al
6,253,324	Field et al

5,918,054	Jury et al
5,740,370	Battersby et al
6,192,477 B1	Corthell
6,170,055 B1	Meyer

Horwitt, Communication Software: 104 Packages to get you on line, 11-1983, Business Computer Systems, volume 2,abstract

Rad, Virus threat bytes computer users, 08-1988, HoustonChronicle.com, start page 6

Brown, Reader Response Reveals e-mail virus hoax, 03-1998, Roanoke Times & World New, start page A5

Cornetto, HTML provides opening for email vandals, 08-1998, CNN.

Microsoft Computer Dictionary

Newton's Telecom dictionary

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Paula W Klimach whose telephone number is (703) 305-8421. The examiner can normally be reached on Mon to Fri 7:15 a.m to 3:45 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gail Hayes can be reached on (703) 305-9711. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 305-8421 for regular communications and (703) 305-8421 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-4832.

Gail Hayes

GAIL HAYES
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

***
December 13, 2002